

Re: Updating DB Locally

Source: <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2004-06/1366.html>

From: Paige (noemail_at_hotmail.com)

Date: 06/22/04

Date: Mon, 21 Jun 2004 23:12:51 GMT

Gordon Burditt wrote:

>>I am the only one that changes the database entries. There is no form
>>info being directly submitted to the db and no one else touches it. So
>>at all times, the db on my computer is the most up-to-date one.
>>
>>I'm sorry but I don't know the difference between overwriting and
>>synchronizing a database.
>
>
> If the "production" database is truly read-only, there is no difference.
>
> If there are modifications to both databases (say, there is a 'last
> used' field changed when a user accesses the web site, or the user
> can change his/her password), then you have to merge the changes
> for true synchronization, and overwriting just wipes out all the
> changes made in the production database from the web and replaces
> them from the master copy. Real synchronization requires intelligent
> information about what fields are updated on which database for
> what purpose. And if the tables are designed poorly, it may be
> impossible to do right at all. Overwriting is always possible
> (assuming the system is up and you don't run out of disk space,
> etc.) but in practice it may be a disaster (lost updates).

But as I said, there is no writing done to the database on the server at all. It does not contain users/passwords. It contains information only and is not used as any part of a login system. The only copy that changes is the one on my computer. When I have finished making those changes, I need a streamlined way to upload them from my computer to the server which is not mine.

> Since your updates seem to be entered in one place only, you might
> consider setting up replication if you are willing to have changes
> made on the test server appear almost instantly on the production
> server (which could ruin its use as a 'test' server, or it might
> just eliminate the chore of propagating updates entirely).

Where would I find out more about this method?

comp.lang.php: Re: Updating DB Locally

>
>>*I only know that I'd like to be able to upload*
>>*the latest db info from my computer to the server and have that one*
>>*overwrite, replace, update (or whatever terminology fits) the one that's*
>>*on the server.*
>
>
> *I'd look at various options of mysqldump. You don't have to delete*
> *and re-create the tables if the schema didn't change (-t, no table*
> *creation info). You can also lock the tables so all the updates*
> *get done, from the point of view of the web server, all at once.*
>
>
>>*As it stands right now, the only way I can accomplish this is to either*
>>*drop the entire server database and reinstall it using the updated copy*
>>*or else I have to edit all the entries in the .sql backup I make to*
>>*delete existing entries and mark the ones I changed as UPDATE instead of*
>>*INSERT. Neither way is particularly appealing, thus my inquiry as to an*
>>*easier way to do this.*
>
>
> *I don't understand why ignoring warnings about inserting duplicate*
> *keys is a problem. I'm presuming that you have UNIQUE keys set up*
> *so re-inserting the same data you already have won't result in*
> *duplicate records. The mysql command-line utility has the -f option,*
> *keep going on SQL error, which will take care of stopping on hitting*
> *an already-existing record.*

I don't have command line access to the server database, only to the copy on my computer. The duplicate entries is a problem because sometimes the info in a specific entry changes (people move, change phone numbers, marry, etc) and I need the new info to overwrite or somehow supplant the old when I upload my changes. The db only has 5 fields and the primary is the auto incrementing ID.

> *mysqldump also permits adding a WHERE clause to the data you dump,*
> *so if there is a time stamp (or some other means you can identify*
> *new records, like an autoincrement sequence number if you only ADD*
> *new records and never change existing ones) on the records, you can*
> *just dump stuff since the last dump and load those. Of course you*
> *have to keep track of this info for the next dump/copy.*

I understand that part. Adding the new fields only is easy enough even using my backup file, open file, edit file, save file, upload file method. It's the fields that changed that are more work. I'll have to read more on mysqldump to see if it's something I can use or adapt.

> *Is the volume of data such that uploading all of it each time is a*
> *problem?*

Well there are over 700 entries in it right now and it's still growing.